



# CLARAty and Challenges of Developing Interoperable Robotic Software

Coupled Layer Architecture for Robotic Autonomy

Issa A.D. Nesnas

Ames Research Center

Carnegie Mellon University

Jet Propulsion Laboratory

University of Minnesota

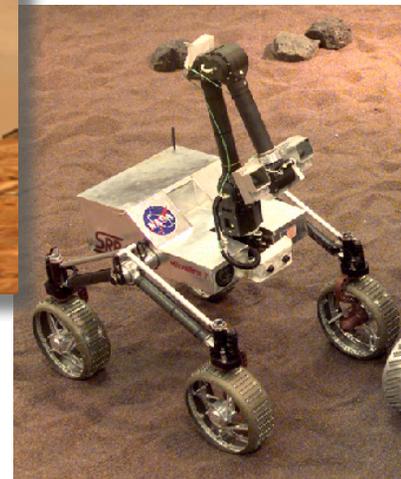
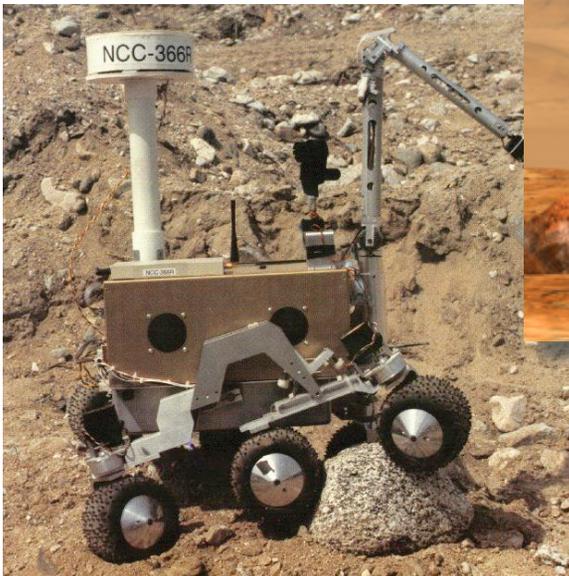
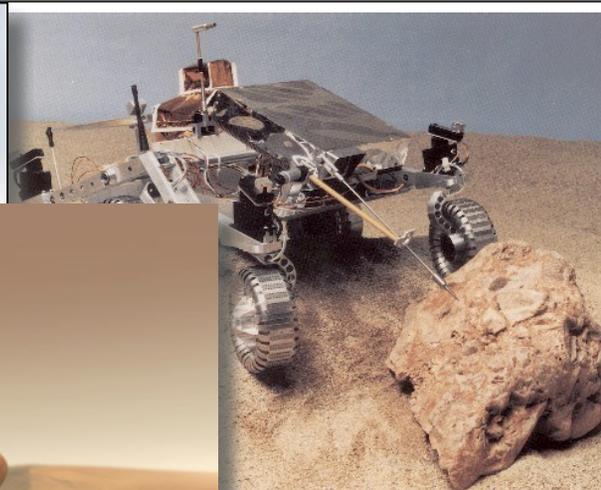
IEEE Aerospace Conference - Big Sky, Montana 2004



# Motivation



- Problem:
  - Difficult to share software across systems
  - Different hardware/software infrastructure
  - No standard protocols and APIs
  - No flexible code base of robotic capabilities
- Approach
  - Unified robotic framework
  - Capture and integrate legacy algorithms
  - Enable faster technology development
  - Operate heterogeneous robots

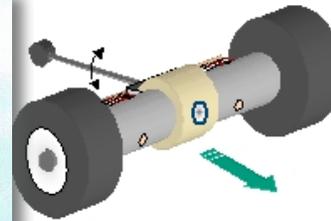




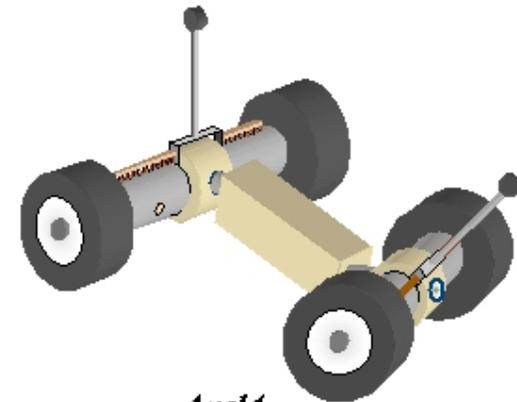
**Custom Rovers**



**Manipulators**



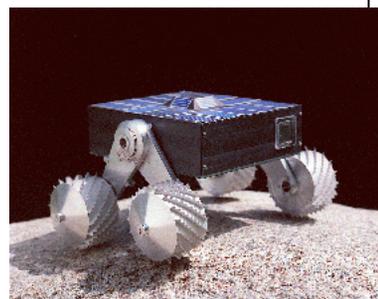
*Axel2*



*Axel4*



**COTS Systems**



**Reconfigurable Robots**



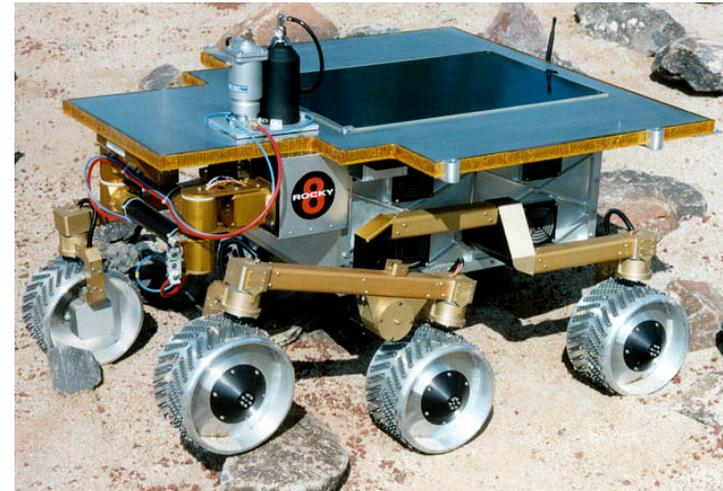


# Challenges in Interoperability

- Mechanisms and Sensors
- Hardware Architecture
- Modular and Reusable Software Components



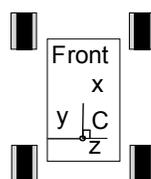
Rocky 7



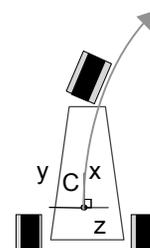
Rocky 8



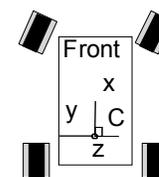
## General flat terrain algorithms and specialized full DOF algorithms



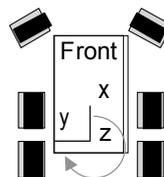
(a)  
Skid Steering  
(no steering wheels)



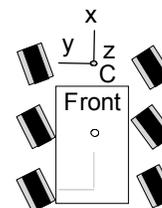
(b)  
Tricycle  
(one steering wheel)



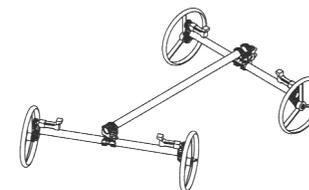
(c)  
Two-wheel steering



(d)  
Partially Steerable  
(e.g. Sojourner,  
Rocky 7)

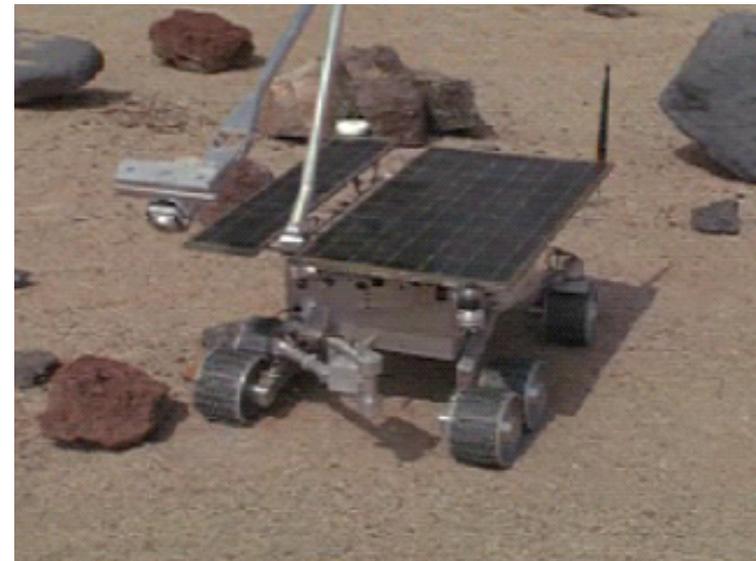
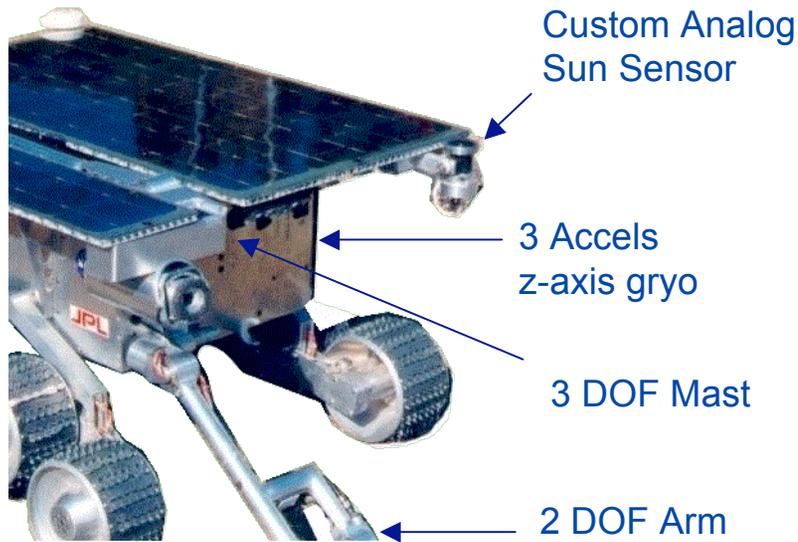


(e)  
All wheel steering  
(e.g. MER, Rocky8,  
Fido, K9)

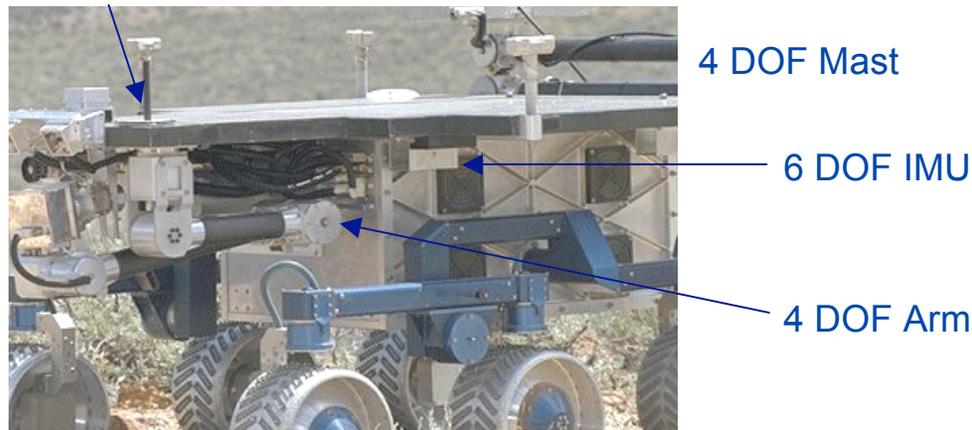


(f)  
Steerable Axle  
(e.g. Hyperion)

• • •



Camera Sun Sensor



- Given different capabilities, how much reuse can be achieved?



# Challenges in Interoperability

- Mechanisms and Sensors
- **Hardware Architecture**
- Modular and Reusable Software Components

# Supported Platforms



*Rocky 8*

VxWorks x86

JPL



*K9*

Linux

x86

Ames



*Rocky 7*

VxWorks ppc

JPL



*FIDO*

VxWorks x86

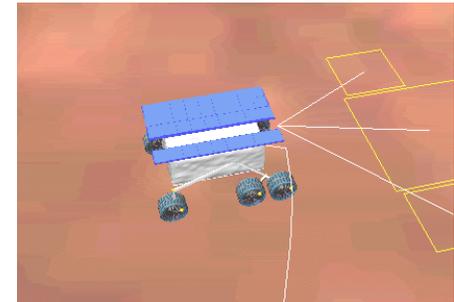
JPL



*ATRV*

Linux x86

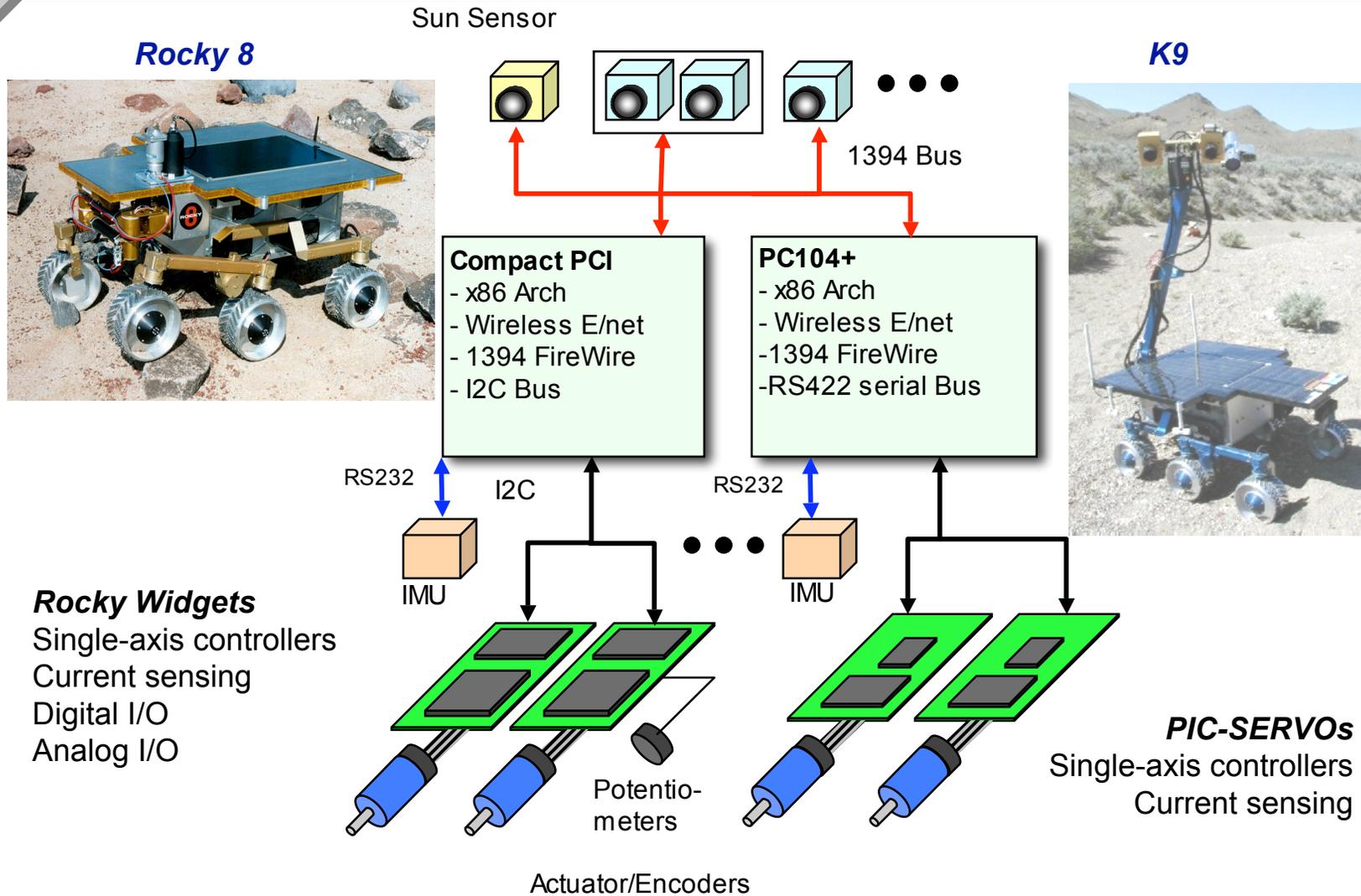
CMU



*ROAMS*

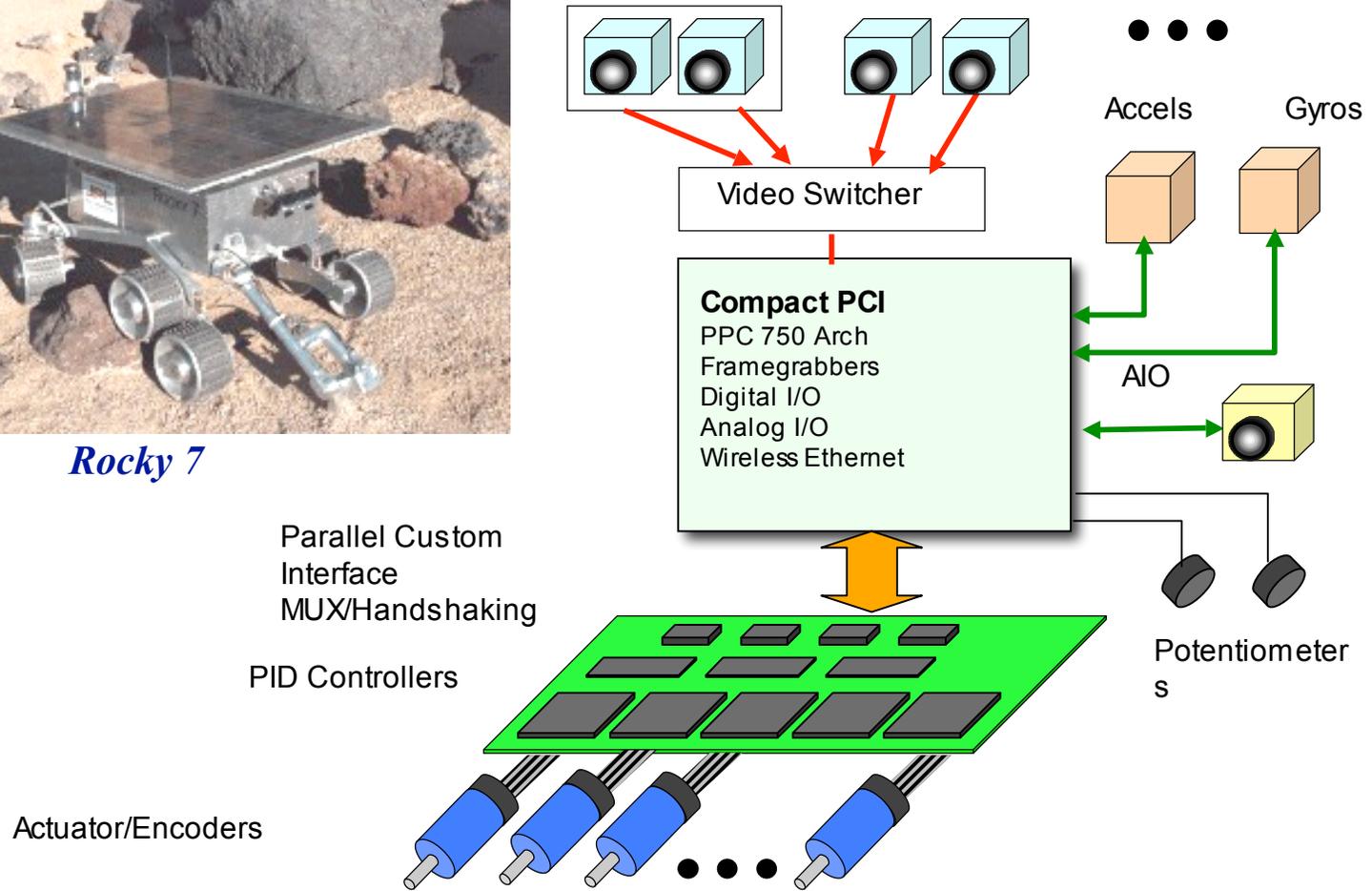
Solaris Linux

JPL





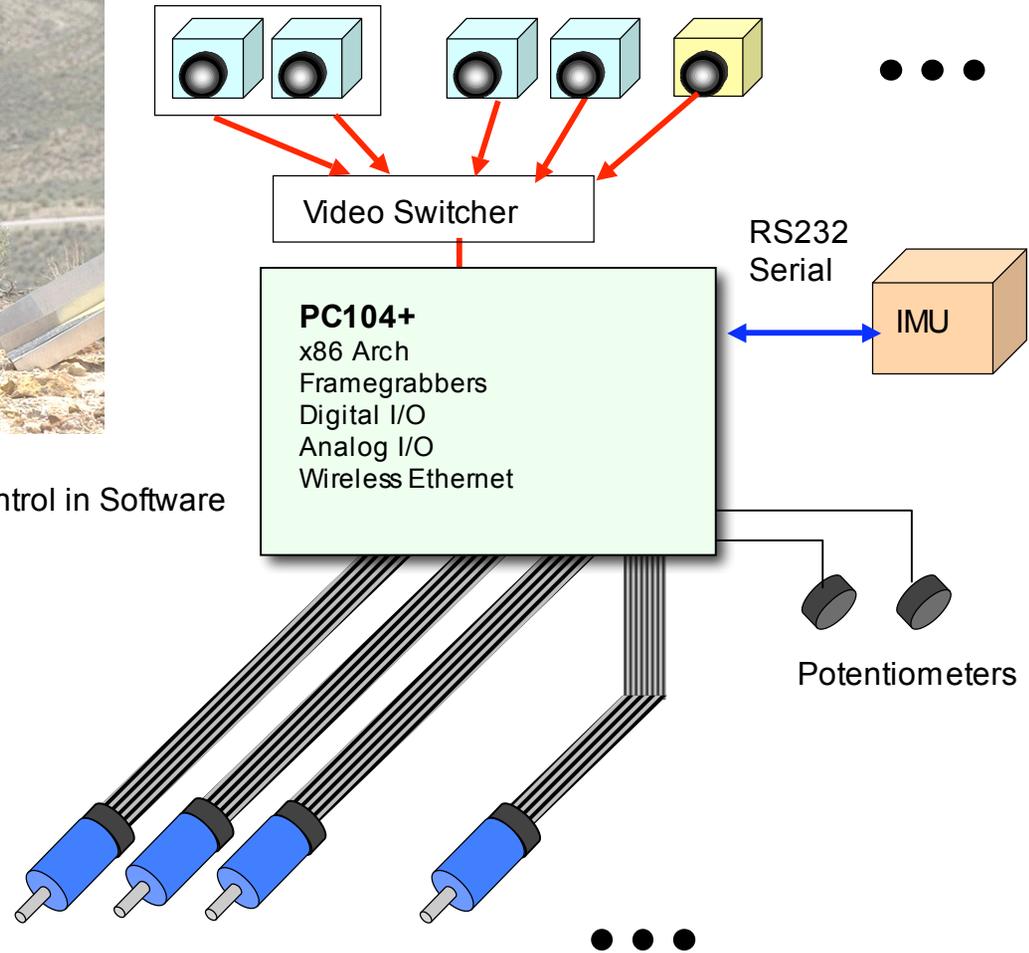
*Rocky 7*

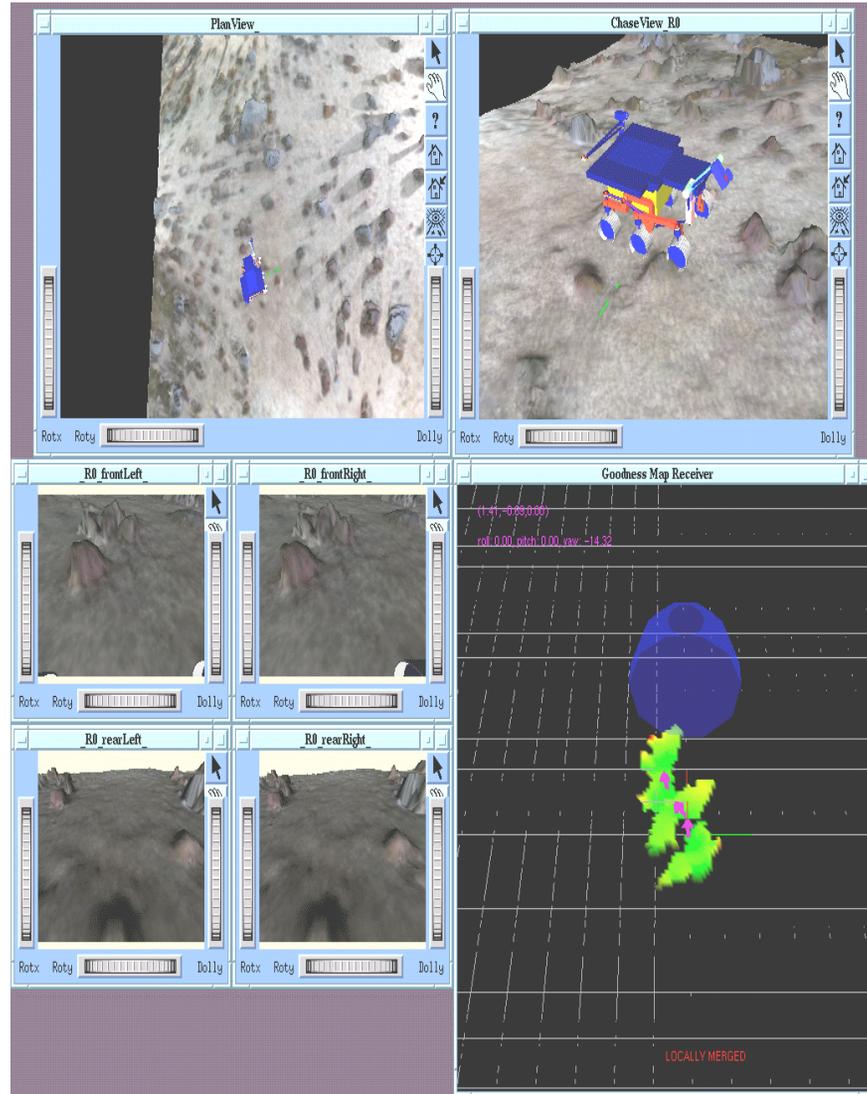




*Fido*

PID Control in Software







# Challenges in Interoperability

- Mechanisms and Sensors
- Hardware Architecture
- Modular and Reusable Software Components

- Different applications have different requirements for system level information
- Develop an understanding of various technologies
- Proper classification working with technology providers
- Integrate multiple such components
- Map to different hardware architectures
- Make trades and balance requirements
- Avoid least common denominator - not acceptable
- Match algorithmic requirement to generic system
- Difficult to predict future algorithmic requirements

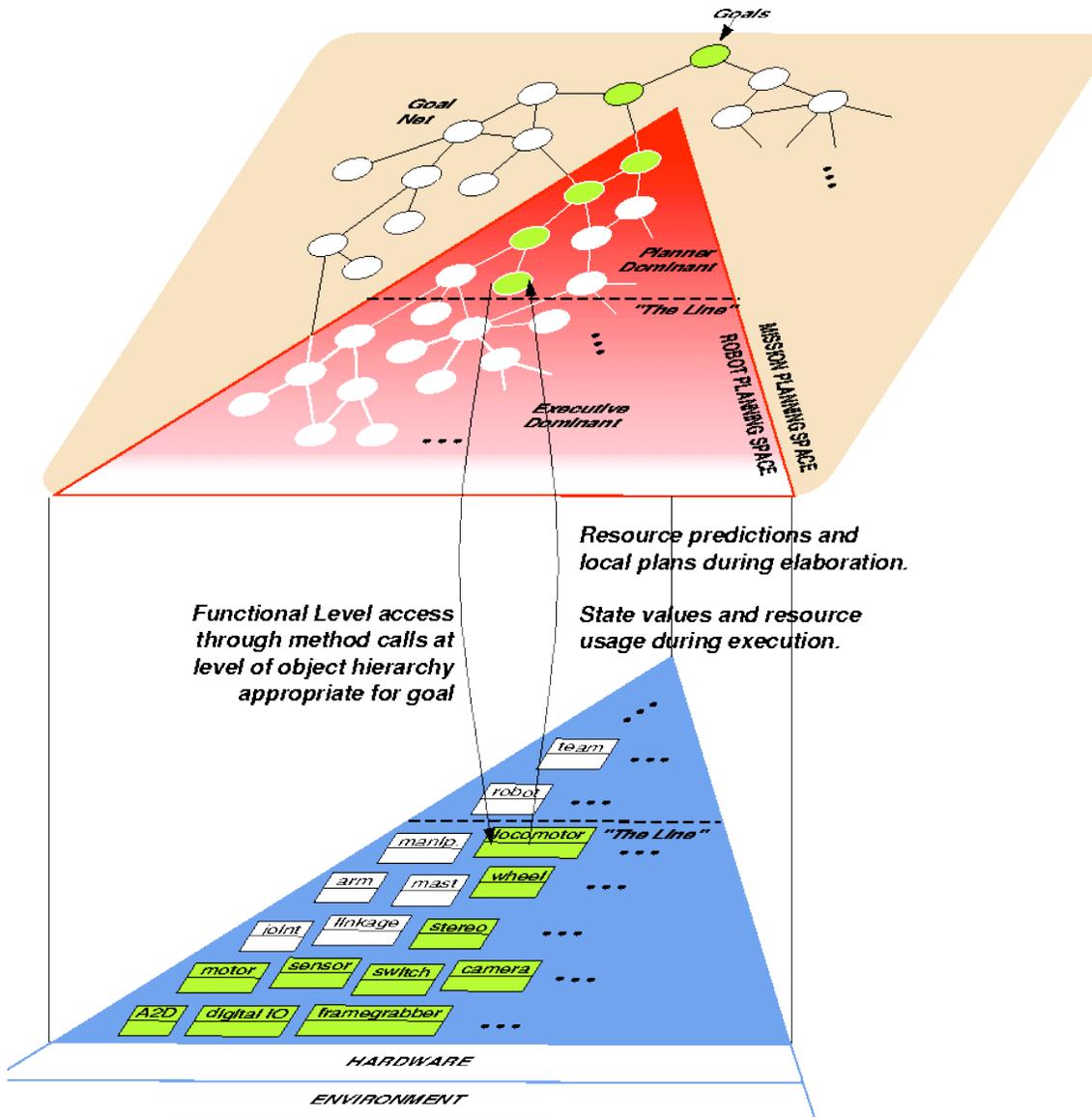


# Technical Approach



# A Two-Layered Architecture

CLARAty = Coupled Layer Architecture for Robotic Autonomy



## **THE DECISION LAYER:**

Declarative model-based  
Global planning

## **INTERFACE:**

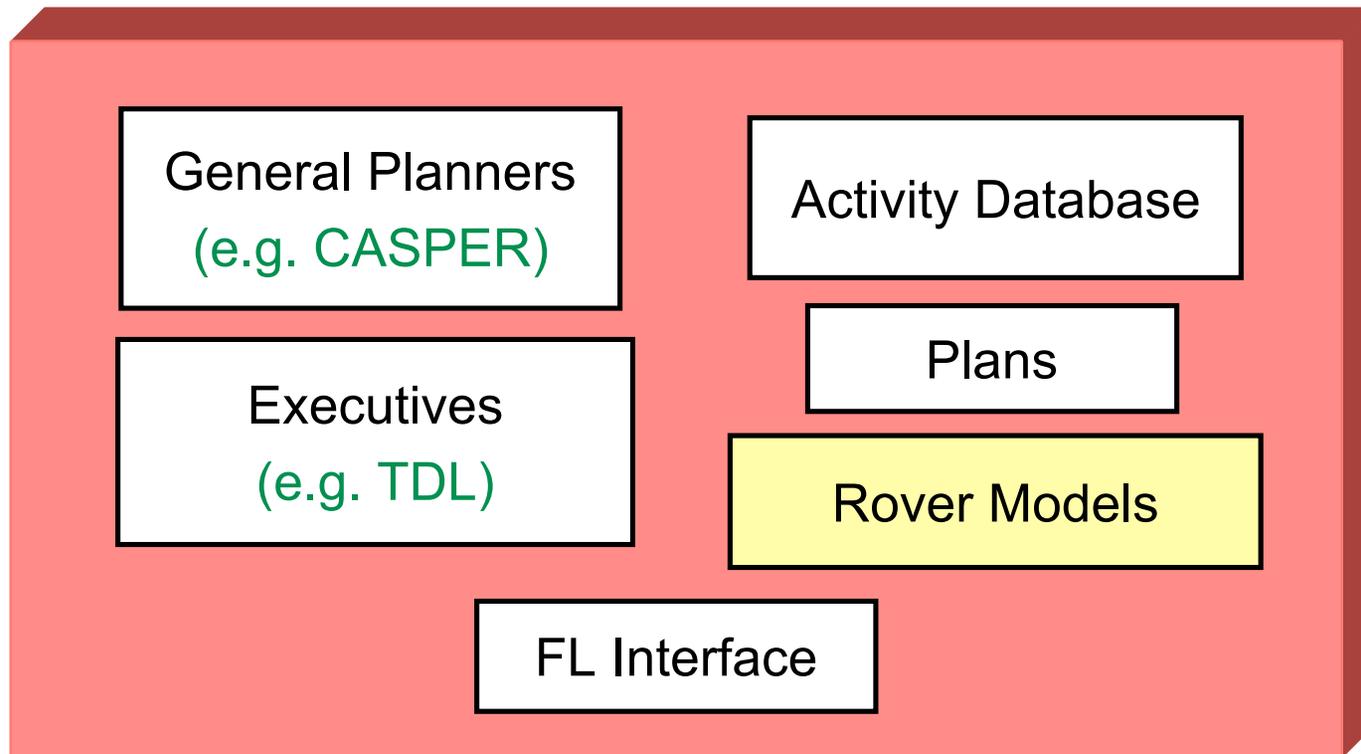
Access to various levels  
Commanding and updates

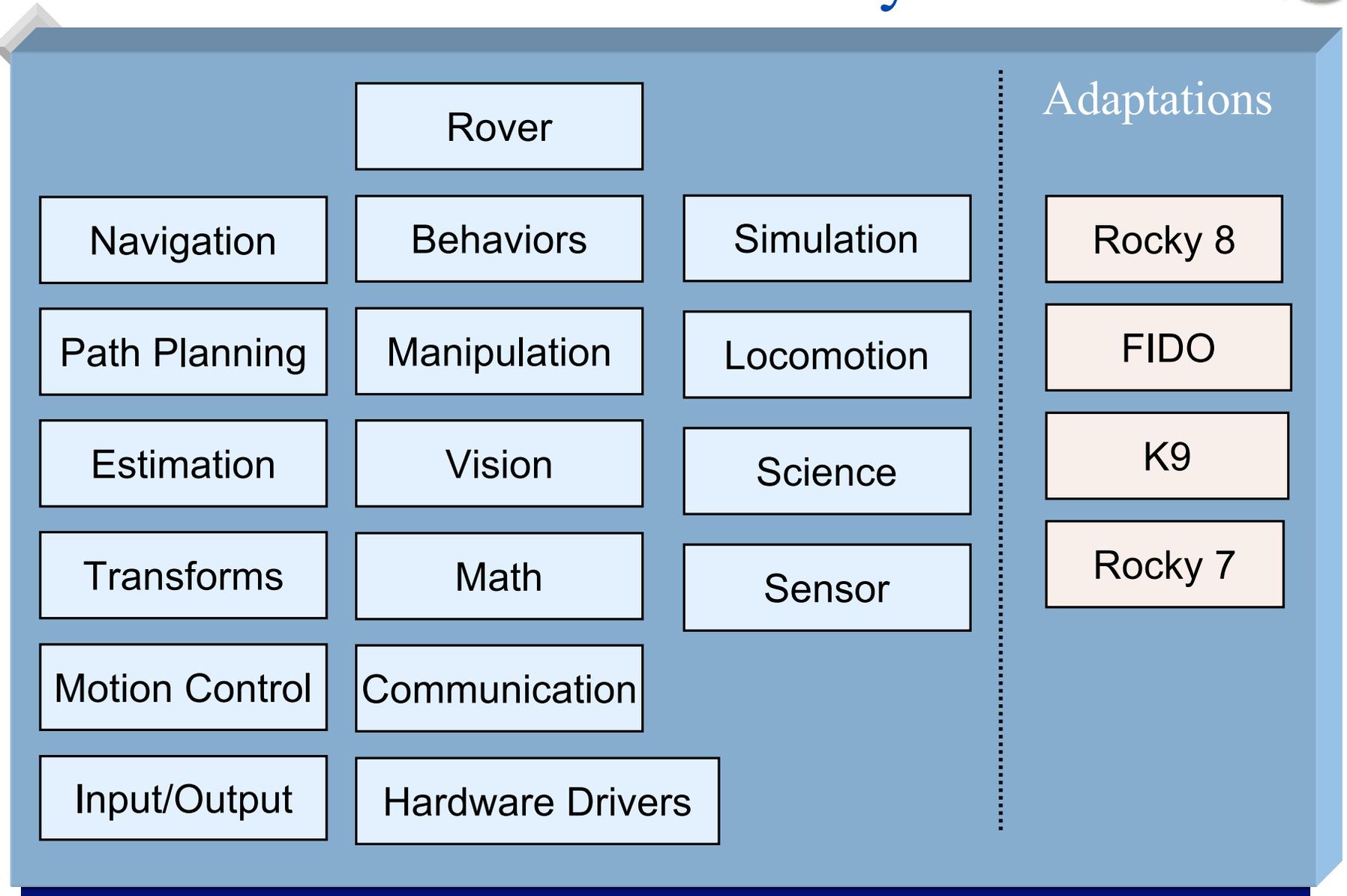
## **THE FUNCTIONAL LAYER:**

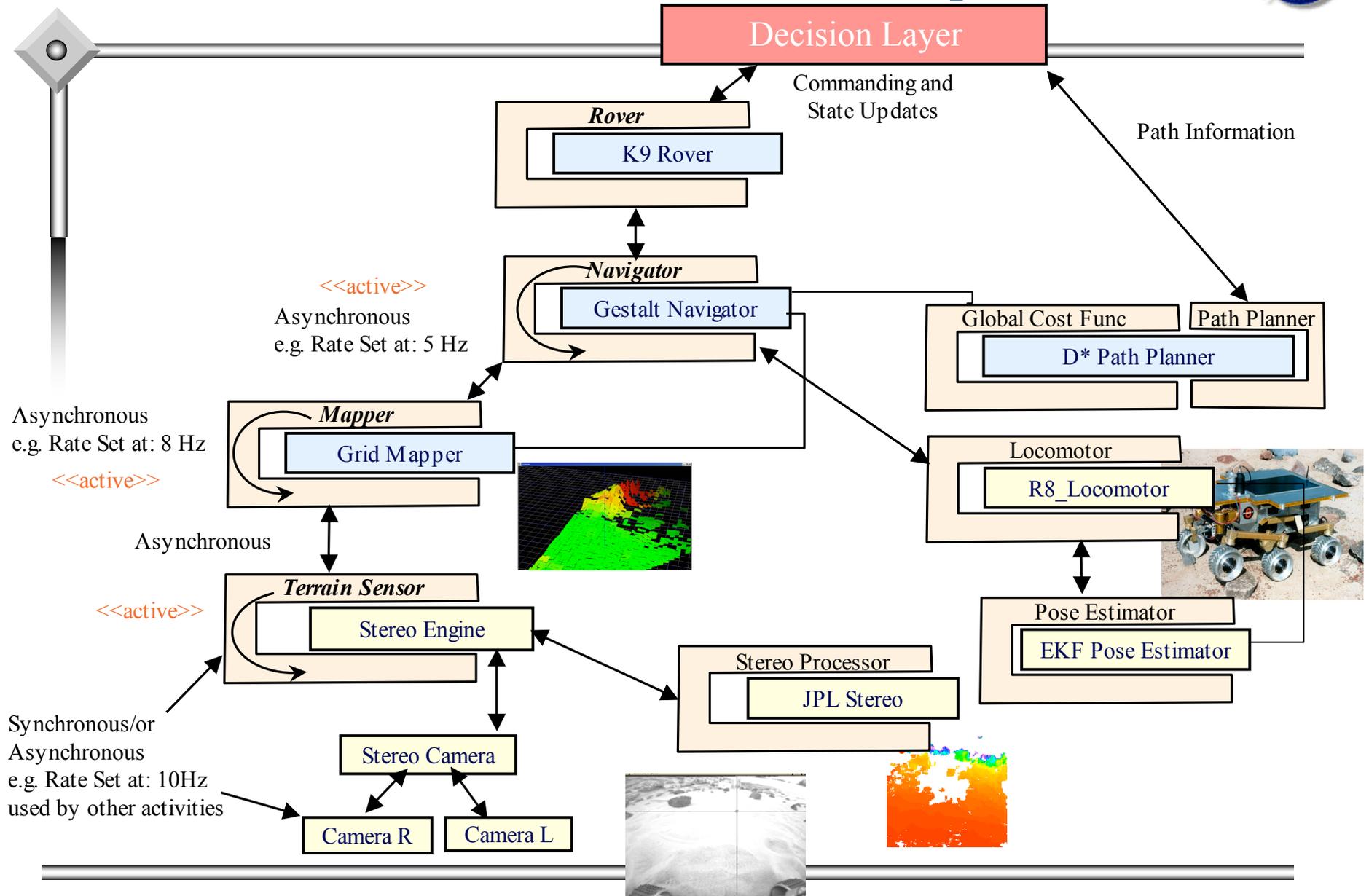
Object-oriented abstractions  
Autonomous behavior  
Basic system functionality

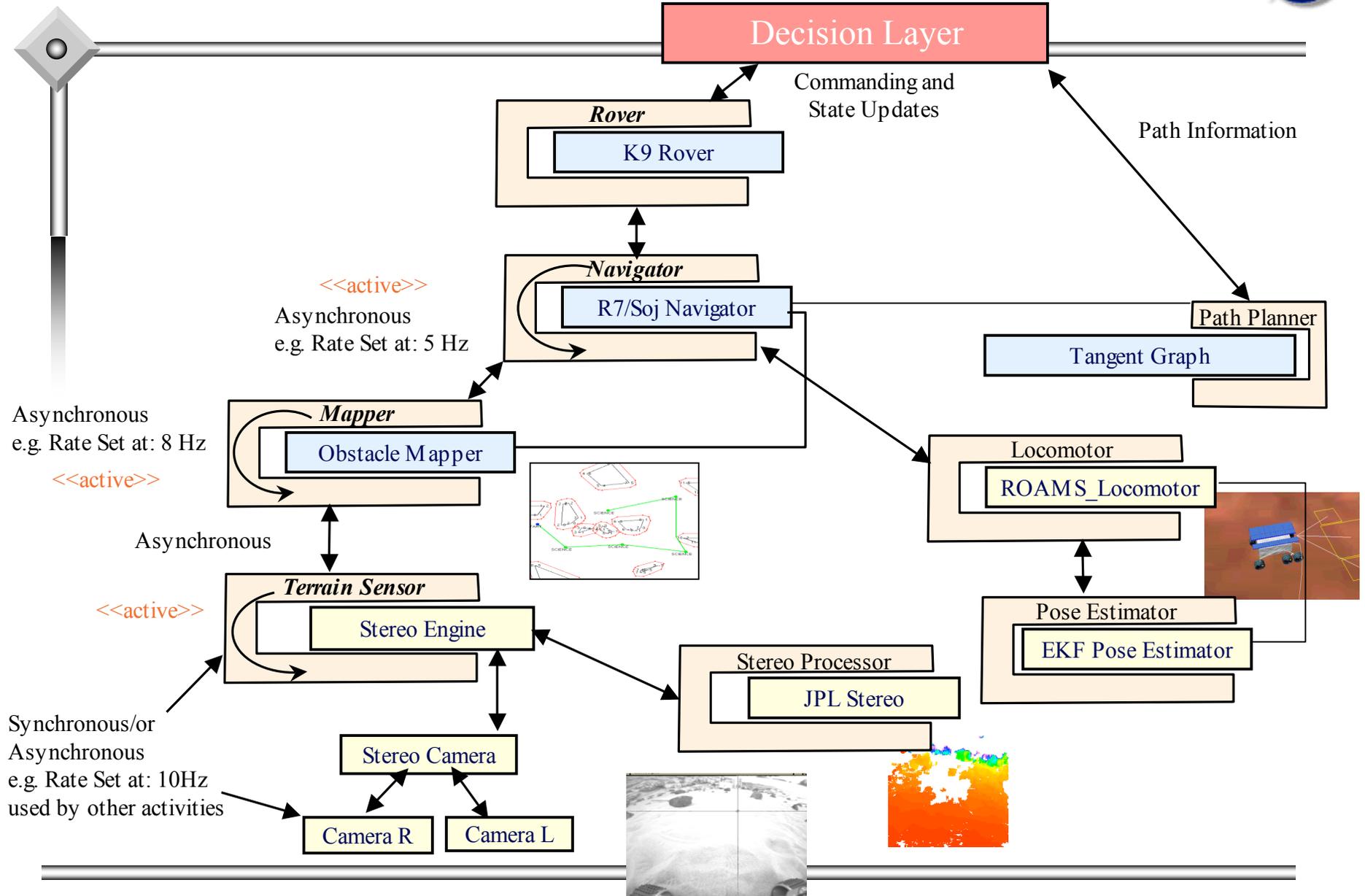
Adaptation to a system

# The Decision Layer



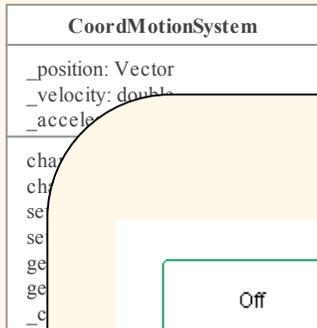




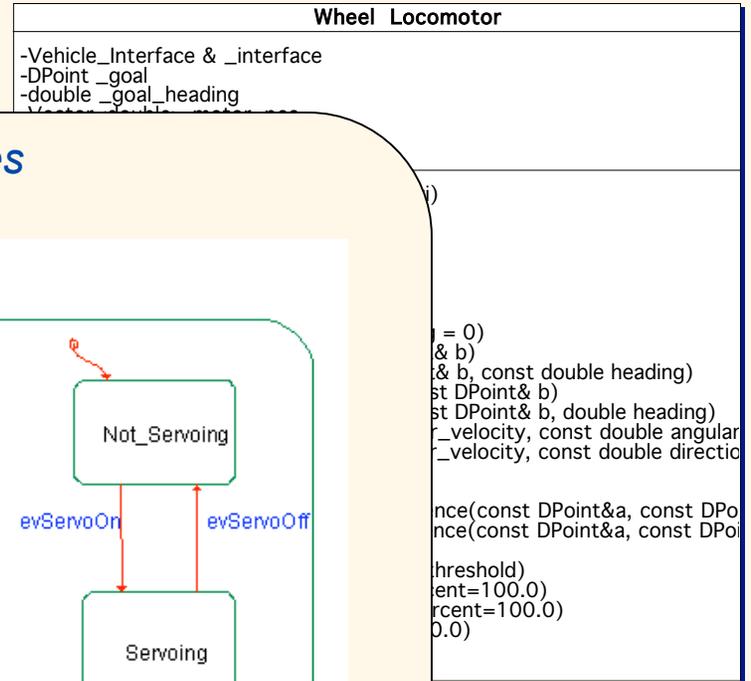


- Data Structure Components
  - Array, Vector, Matrix, Map, Container, LinkedList, Bit
  - Image, Message, Resource
- Generic Physical Components (GPC)
  - Device, Locomotor, Manipulator, Spectrometer
- Specialized Physical Components (SPC)
  - K9\_Locomotor, K9\_Arm, R8\_Mast, FW\_Camera
- Generic Functional Components (GFC)
  - ObjectFinder, VisualNavigator, Stereovision, Localizer
- Specialized Functional Components (SFC)
  - ARC\_Stereovision, JPL\_Visual\_Odometer

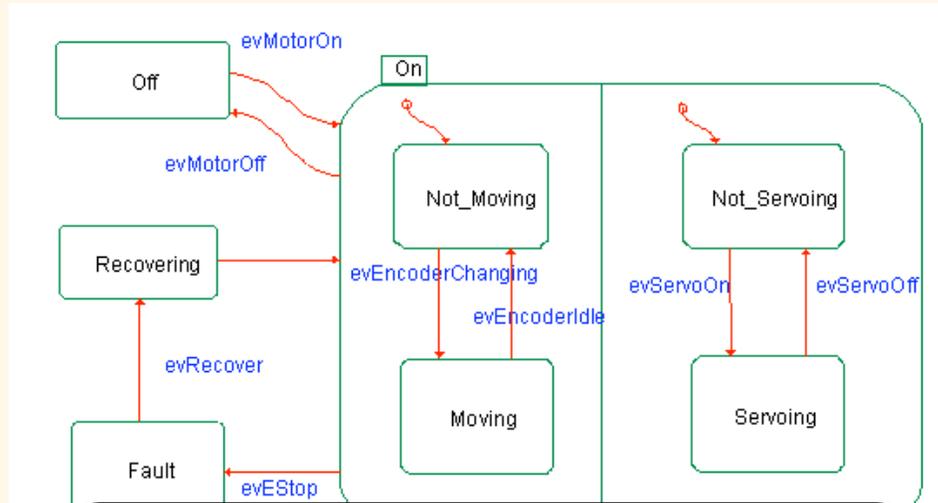
## Abstractions



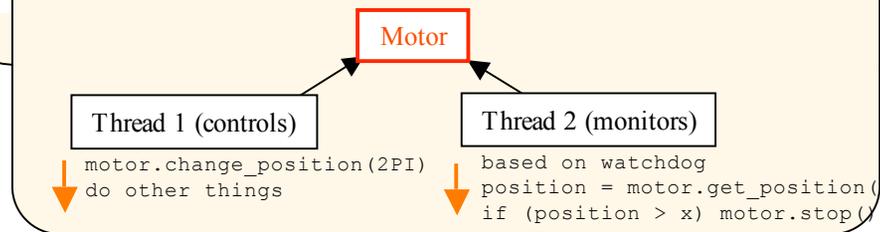
## APIs and Behaviors

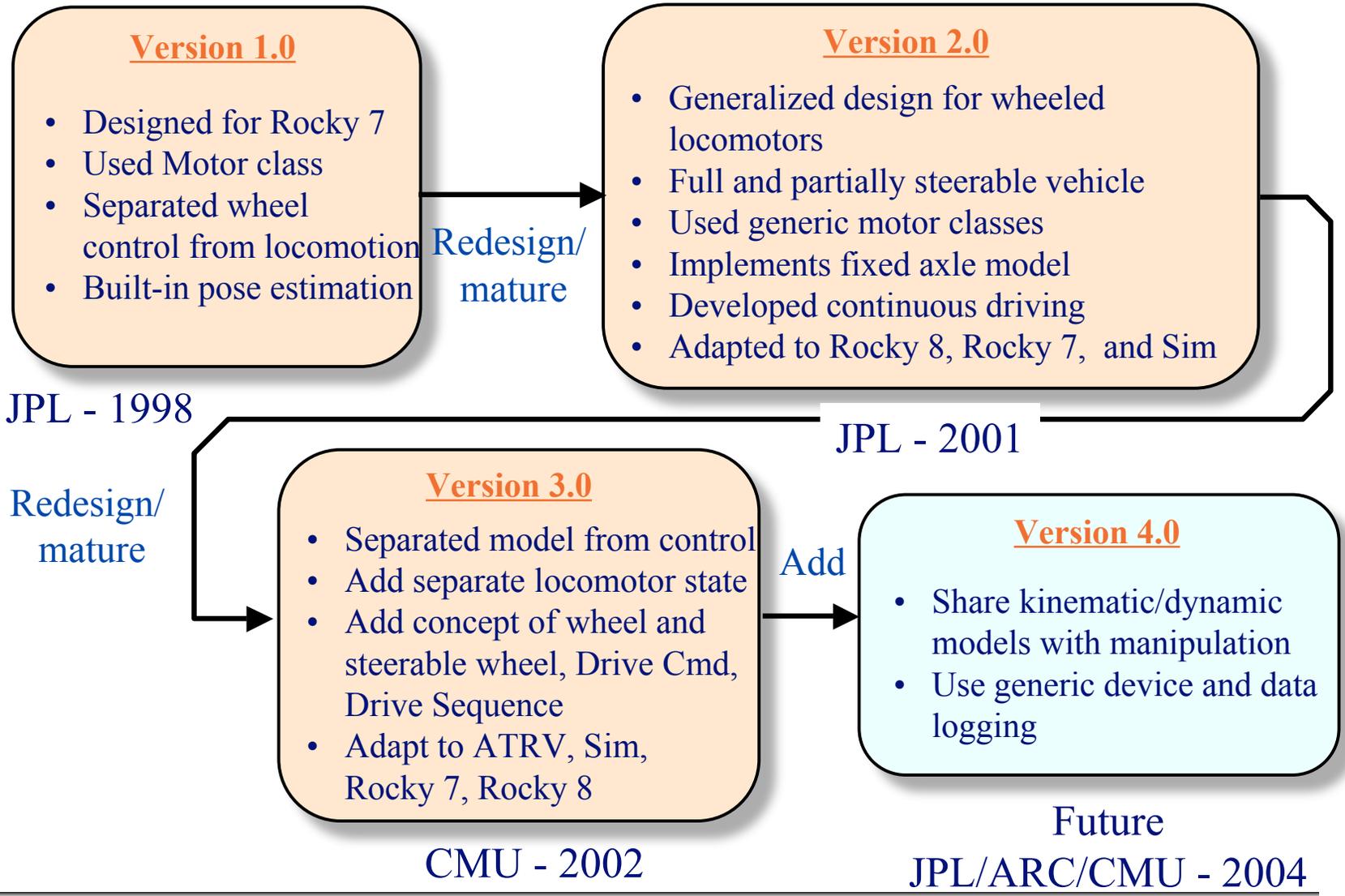


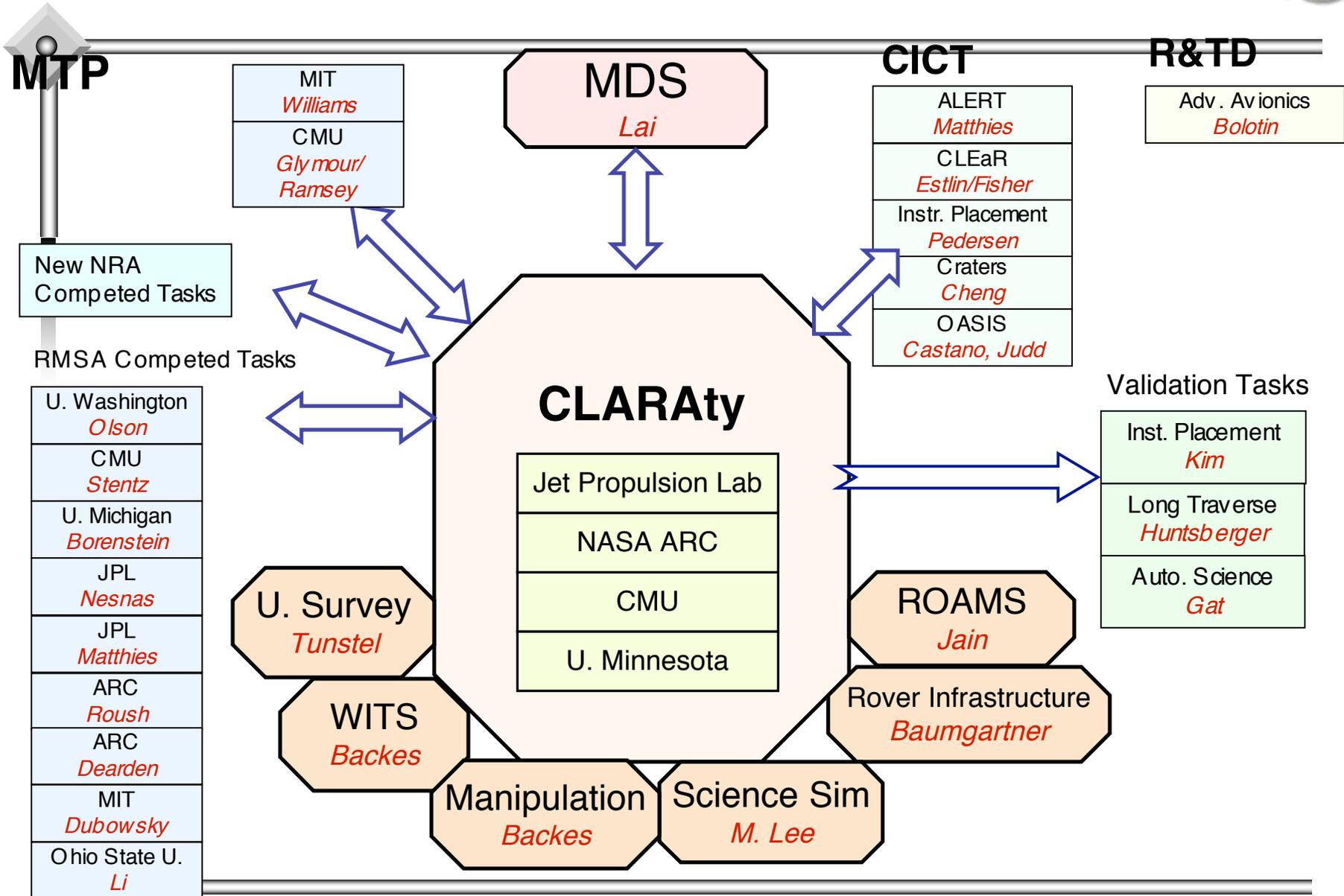
## State Machines



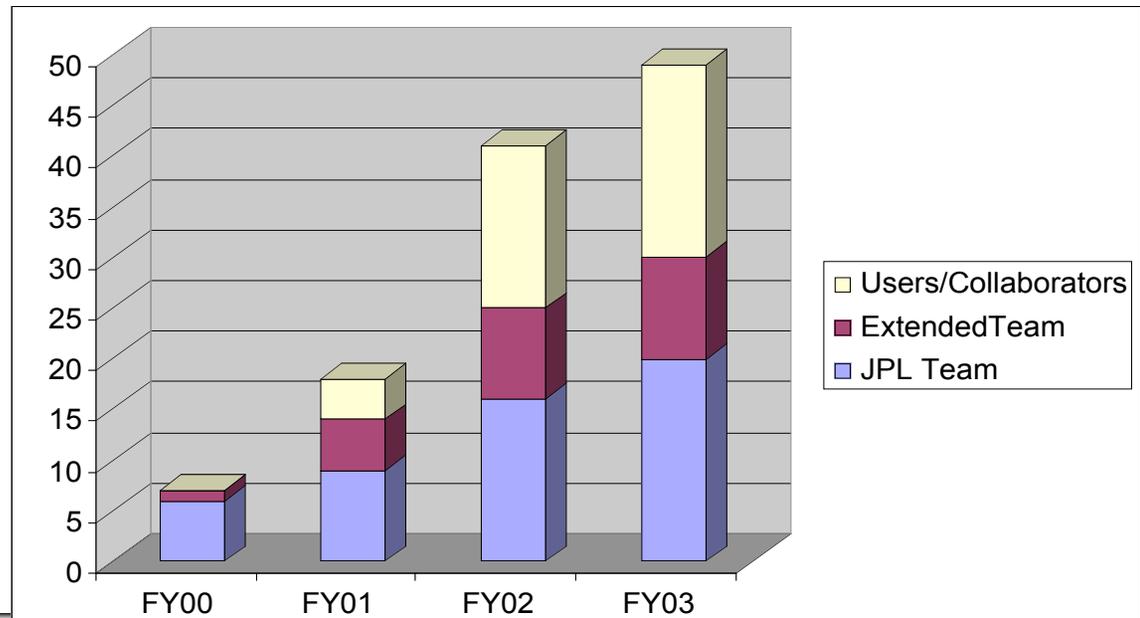
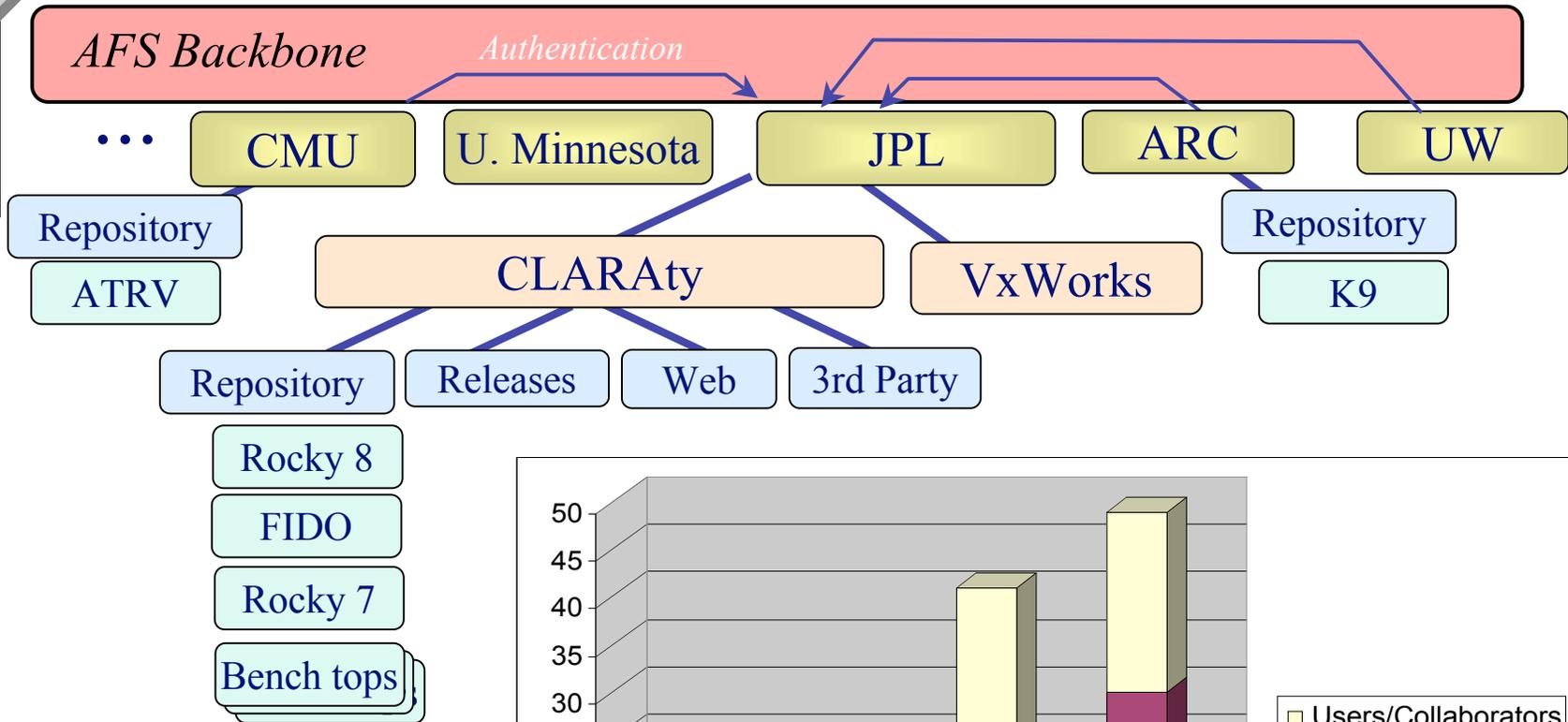
## Runtime Models



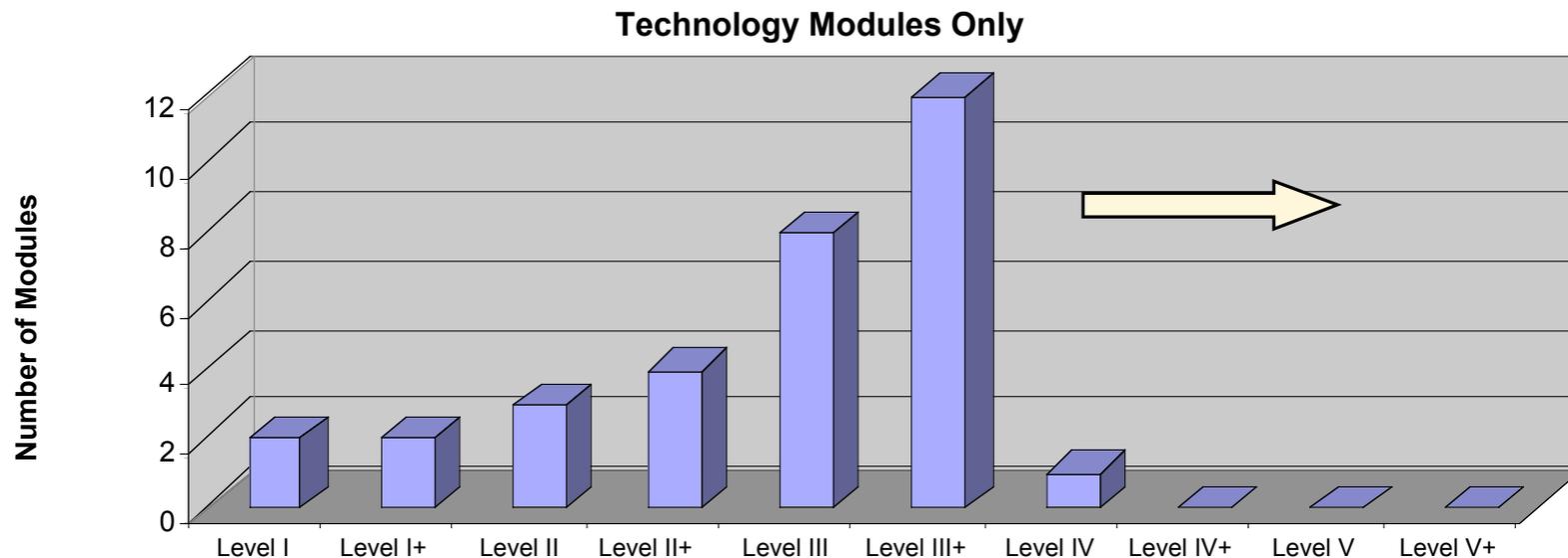


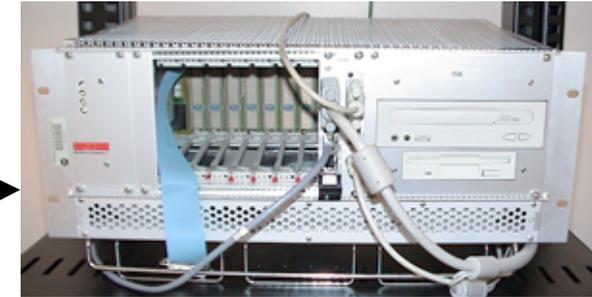


- **Level I - Deposited**
  - Code exists in CLARAty repository - all Intellectual Properties items cleared
  - Compiles as a standalone application - no dependencies to other modules
  - Has test programs and user documentation for getting started
- **Level II - Encapsulated**
  - Integrated with other CLARAty modules
  - Uses CLARAty components to interact with rover
  - Does not support a CLARAty API
  - Runs on at least one robot platform
- **Level III - Integrated**
  - Conforms to a generic CLARAty API (or parent class)
  - Has no unsupported 3rd party dependencies
  - Runs on all applicable rover platforms
- **Level IV - Refactored and Reviewed**
  - Software reviewed by committee to ensure internal/external consistency
  - Uses all applicable CLARAty classes
  - Internally conforms to CLARAty conventions and coding standards
- **Level + - Reused**
  - Re-used by other modules in CLARAty - dependent module
  - Provides access to all internal data products

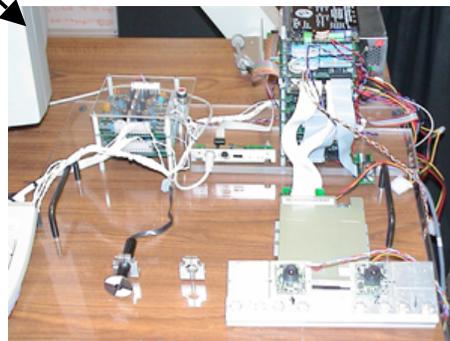
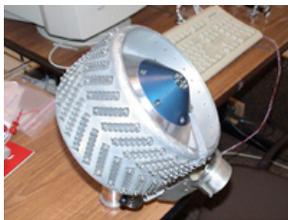


- About 300 modules in Repository – goal is to limit modules
- About 500,000 lines of C++ code – revise and reduce
- Five adaptations to the following rovers:
  - Rocky 8, FIDO, Rocky 7
  - ATRV
  - K9
- Most technology modules are at Level II and Level III
- None are at Level IV or Level V (fully compliant, documented, and formally reviewed)

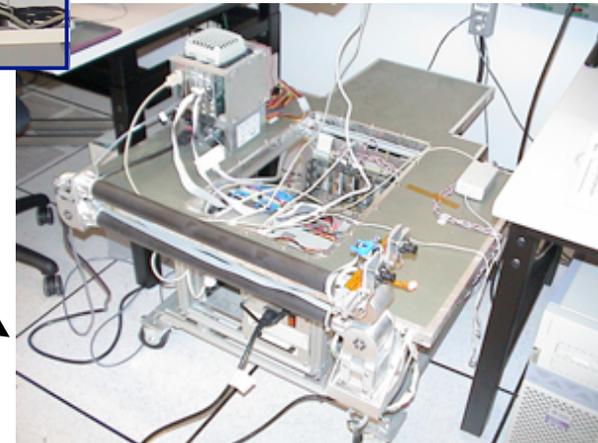




Rocky 8 Bench top



FIDO Benchtop



Dexter Manipulators  
Rocky 7 Bench top

- Use abstraction to master complexity
- Encapsulate and abstract hardware variations
- Provide multi-level access through Decision Layer for fault diagnosis and recovery
- Use domain expertise to guide design
- Make all assumptions explicit
- Stabilize external interfaces rapidly
- Document processes and products well
- Avoid over-generalization - define scope
- Encapsulate system specific runtime models
- Do not compromise performance - least common denominator solutions are unacceptable in hw/sw interactions
- Standardize Hardware





# Acknowledgements

CLARAty Team (multi-center)

## Jet Propulsion Laboratory

- ROAMS/Darts Team
- CLEaR Team
- Instrument Simulation Team
- Machine Vision Team
- FIDO Team



## Ames Research Center

- K9 Team



## Carnegie Mellon University



# CLARAty Team



## • NASA Ames Research Center

- Maria Bualat
- Sal Desiano
- Clay Kunz (*Data Structure Lead*)
- Eric Park
- Randy Sargent
- Anne Wright (*Cog-E & Core lead*)

## • Carnegie Mellon University

- David Apelfaum
- Reid Simmons (*Navigation lead*)
- Chris Urmson
- David Wettergreen

## • University of Minnesota

- Stergios Roumeliotis

## • Jet Propulsion Laboratory

- Max Bajracharya (34) (*Cog-E & Vision lead*)
- Edward Barlow (34)
- Antonio Diaz Calderon (34)
- Caroline Chouinard (36)
- Gene Chalfant (34)
- Tara Estlin (36) (*Deputy Manager & Decision Layer lead*)
- Erann Gat (36)
- Dan Gaines (36) (*Estimation Lead*)
- Mehran Gangianpour (34)
- Won Soo Kim (34) (*Motion lead*)
- Michael Mossey (31)
- Issa A.D. Nesnas (34) (*Task Manager*)
- Richard Petras (34) (*Adaptation lead*)
- Marsette Vona (34)
- Barry Werger (34)

## • OphirTech

- Hari Das Nayar



*Thank you for your Attention*

